AMENDMENTS TO THE CLAIMS

1. (Canceled)

2. (Currently amended) A method as defined in claim[[1]] 32, further comprising decrementing a loop count in the architectural entry in the register file ~~in response to~~ when the loop bottom instruction ~~being committed~~ has completed execution in the pipeline.

3-4. (Canceled)

5. (Currently amended) A method as defined in claim[[1]] 31, further comprising generating a current pointer to the current entry in the register file and generating an architectural pointer to the architectural entry in the register file.

6. (Original) A method as defined in claim 5, further comprising incrementing the current pointer to a second entry in the register file in response to decoding of a second loop setup instruction and writing in the second entry loop control parameters represented in the second loop setup instruction.

7. (Currently amended) A method as defined in claim 6, further comprising incrementing the architectural pointer to the second entry in the register file ~~in response to~~ when the second loop setup instruction ~~being committed~~ has completed execution in the pipeline.

8. (Original) A method as defined in claim 6, further comprising moving the current pointer to a location of the architectural pointer in response to an interrupt or a pipeline abort.

9.-10. (Canceled)

11. (Currently amended) A method as defined in claim[[1]] 31, further comprising writing a temporary loop count in a temporary loop count register and decrementing the temporary loop count on each loop bottom match.

12. (Original) A method as defined in claim 11, further comprising exiting the program loop when the temporary loop count has decremented to zero.

13. (Currently amended) A method as defined in claim[[1]] 31, further comprising stalling a loop setup instruction when the register file does not have an available entry.

14. (Currently amended) A method as defined in claim[[1]] 31, wherein instructions are issued without sending the loop control parameters down the pipeline.

15-17. (Canceled)

18. (Currently amended) Apparatus as defined in claim[[17]] 34, wherein the controller further comprises means for ~~issuing instructions of the program loop according to the loop control parameters in the current entry in the register file,~~ sending a loop bottom indicator down the pipeline with a loop bottom instruction, and means for decrementing a loop count in the architectural entry in the register file ~~in response to~~ when the loop bottom instruction ~~being committed~~ has completed execution in the pipeline.

19. (Currently amended) Apparatus as defined in claim 18, wherein the controller further comprises means for marking a second entry in the register file as the current entry in response to decoding of a second loop setup instruction and for writing in the second entry loop control parameters represented in the second loop setup instruction, and means for marking the second entry in the register file as the architectural entry ~~in response to~~ when the second loop setup instruction ~~being committed~~ has completed execution in the pipeline.

20-21. (Canceled)

22. (Currently amended) Apparatus as defined in claim[[17]] 34, further comprising a temporary loop count register for holding a temporary loop count, wherein the controller further comprises means for decrementing the temporary loop count on each loop bottom match.

23. (Original) Apparatus as defined in claim 22, wherein the controller further comprises means for exiting the program loop when the temporary loop count has decremented to zero.

24. (Original) Apparatus as defined in claim 23, wherein the controller further comprises means for stalling when a loop setup instruction is decoded and the register set does not have an available entry.

25. (Currently amended) Apparatus as defined in claim[[17]] 34, wherein the controller is configured for operation without sending the loop control parameters down the pipeline.

26. (Canceled)

27. (Original) Apparatus as defined in claim 18, wherein the controller includes means for generating a current pointer for marking the current entry in the register file and for generating an architectural pointer for marking the architectural entry in the register file.

28. (Original) Apparatus as defined in claim 27, wherein the controller includes means for incrementing the current pointer in response to decoding of a loop setup instruction.

29. (Currently amended) Apparatus as defined in claim 28, wherein the controller includes means for incrementing the architectural pointer ~~in response to~~ when a loop setup instruction ~~being committed~~ has completed execution in the pipeline.

30. (Original) Apparatus as defined in claim 27, wherein the controller includes means for moving the current pointer to a location of the architectural pointer in response to an interrupt or a pipeline abort.

31. (New) A method for issuing instructions in a processor having a pipeline, comprising:

(a) providing a loop buffer for holding program loop instructions and a register file having at least three entries for holding speculative and architectural loop control parameters, wherein each entry in the register file comprises a loop top register for holding a loop top address, a loop bottom register for holding a loop bottom address and a loop count register for holding a loop count;

(b) in response to decoding of a first loop setup instruction, marking a first entry in the register file as a current entry and writing in the first entry loop control parameters represented in the first loop setup instructions;

(c) sending the first loop setup instruction down the pipeline;

(d) issuing loop instructions corresponding to the first loop setup instruction speculatively after the loop control parameters are written in the current entry in the register file and before the first loop setup instruction has completed execution in the pipeline, wherein the current entry in the register file contains speculative copies of the loop control parameters; and

(e) marking the current entry in the register file as an architectural entry when the first loop setup instruction has completed execution in the pipeline.

32. (New) A method as defined in claim 31, further comprising:

(f) sending a loop bottom indicator down the pipeline with a loop bottom instruction.

33. (New) A method as defined in claim 32, further comprising:

(g) selecting the loop top address of the current entry from the loop top addresses in the register file, comparing a current instruction address with the selected loop top address to determine a loop top match, selecting the loop bottom address of the current entry from the loop bottom addresses in the register file, and comparing the current instruction address with the selected loop bottom address to determine a loop bottom match.

34. (New) Apparatus for issuing instructions in a processor having a pipeline, comprising:

a loop buffer for holding program loop instructions;

a register file having at least three entries for holding speculative and architectural loop control parameters, wherein each entry in the register file comprises a loop top register for holding a loop top address, a loop bottom register for holding a loop bottom address and a loop count register for holding a loop count; and

a controller including means for marking a first entry in the register file as a current entry in response to decoding of a first loop setup instruction and for writing in the first entry loop control parameters represented in the first loop setup instruction, means for sending the first loop setup instruction down the pipeline, means for issuing loop instructions corresponding to the first loop setup instruction speculatively after the loop control parameters are written in the current entry in the register file and before the first loop setup instruction has completed execution in the pipeline, wherein the current entry in the register file contains speculative copies of the loop control parameters, and means for marking the current entry in the register file as an architectural entry when the first loop setup instruction has completed execution in the pipeline.

35. (New) Apparatus as defined in claim 34, further comprising a loop top selector for selecting the loop top address of the current entry from the loop top addresses in the register file, a loop top comparator for comparing a current instruction address with the selected loop top address to determine a loop top match, a loop bottom selector for selecting the loop bottom

address of the current entry from the loop bottom addresses in the register file, and a loop bottom comparator for comparing the current instruction address with the selected loop bottom address to determine a loop bottom match.